

Типовой расчет по дисциплине “Базы данных”

Задание выполнил:

Студент: Шестёркин В.Д.

Группа: А-08-17

Проверил:

Преподаватель:

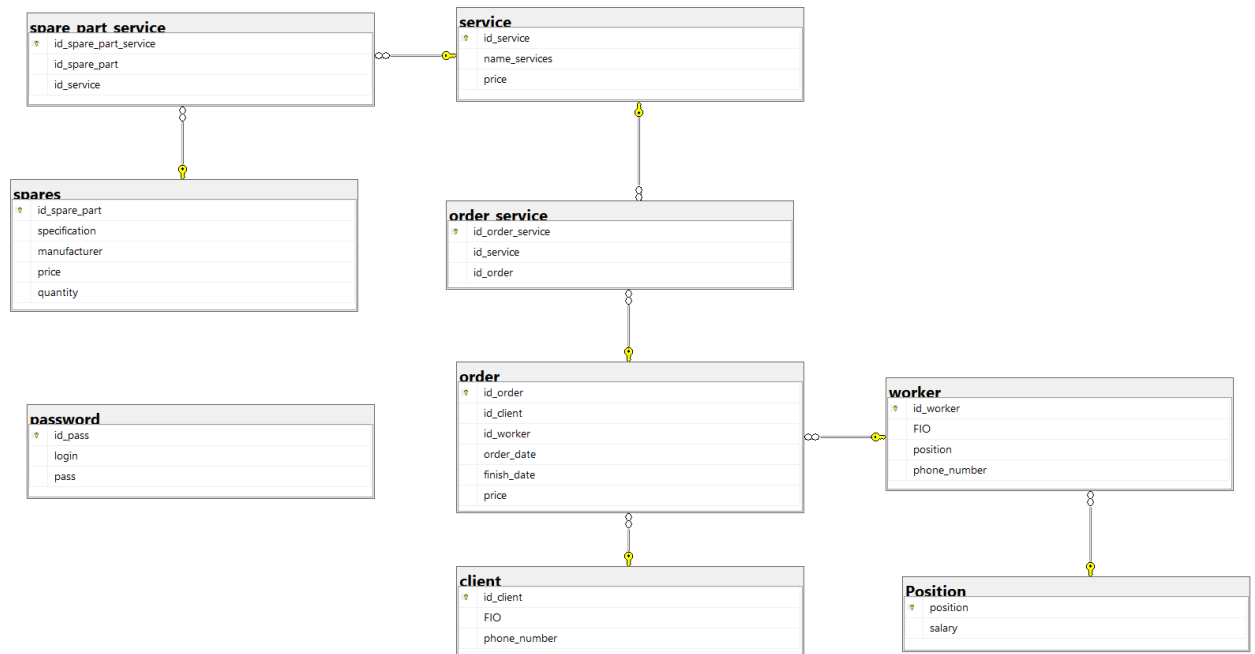
Дата:

Оценка:

1. Проектирование базы данных

Для удобства проектирования все основные ключи являются индексами (с автоинкрементом).

ER-диаграмма:



2. Создание базы данных

В качестве реляционной СУБД в данной работе использовался Microsoft SQL Server 2016 с применением языка запросов Transact-SQL. Проектирование базы данных производилось в среде Microsoft SQL Server Management Studio 2018.

Таблицы	Представления	Процедуры	
<ul style="list-style-type: none"> dbo.client dbo.order dbo.order_service dbo.password dbo.Position dbo.service dbo.spare_part_service dbo.spares dbo.worker 	<ul style="list-style-type: none"> dbo.view_accountant dbo.view_procurement dbo.view_repairer 	<ul style="list-style-type: none"> dbo.add_client dbo.add_order dbo.add_order_service dbo.add_service dbo.add_spare_part_service dbo.add_spares dbo.add_worker dbo.del_client dbo.del_order dbo.del_order_service dbo.del_service dbo.del_spare_part_service dbo.del_spares dbo.del_worker dbo.up_client dbo.up_position dbo.up_spares 	
Табличные функции	Скалярные функции	Триггеры	
<ul style="list-style-type: none"> dbo.premiya dbo.price_of 	<ul style="list-style-type: none"> dbo.stringCLR dbo.TOP_spare 	<ul style="list-style-type: none"> dbo.order Столбцы Ключи Ограничения Триггеры PlusZP 	<ul style="list-style-type: none"> dbo.order_service Столбцы Ключи Ограничения Триггеры Plus_price

3. Код некоторых процедур, функций, триггера и представлений.

Представления:

Представление для менеджера по продажам:

```
create view [dbo].[view_procurement] as
select dbo.[order].id_order as 'Номер заказа',
dbo.[order].order_date as 'Дата заказа',dbo.service.name_services as 'Услуги',
dbo.spares.id_spare_part as 'Номер детали',dbo.spares.quantity as 'Количество деталей'
from dbo.[order] join order_service on
(dbo.[order].id_order=dbo.order_service.id_order)
join dbo.service on(dbo.order_service.id_service=dbo.service.id_service)
join dbo.spare_part_service on (dbo.service.id_service=dbo.spare_part_service.id_service)
join dbo.spares on(dbo.spare_part_service.id_spare_part=dbo.spares.id_spare_part)
order by dbo.[order].order_date desc offset 0 rows;
```

Представление для ремонтников:

```
create view [dbo].[view_repairer] as
select dbo.[order].id_order as 'Номер заказа',dbo.[order].id_worker as 'Номер
сотрудника',
dbo.worker.FIO as 'Имя сотрудника',dbo.[order].order_date as 'Дата заказа' ,
dbo.[order].finish_date as 'Дата выполнения',dbo.service.name_services as 'Услуги',
dbo.spares.id_spare_part as 'Номер детали',dbo.spares.specification as 'Описание детали'
--dbo.spares.manufacturer as 'Производитель'--,dbo.spares.quantity as 'Количество деталей
на складе'
from dbo.[order] join dbo.worker on (dbo.[order].id_worker = dbo.worker.id_worker)
join order_service on (dbo.[order].id_order=dbo.order_service.id_order)
join dbo.service on(dbo.order_service.id_service=dbo.service.id_service)
join dbo.spare_part_service on (dbo.service.id_service=dbo.spare_part_service.id_service)
join dbo.spares on(dbo.spare_part_service.id_spare_part=dbo.spares.id_spare_part)
order by dbo.[order].order_date desc offset 0 rows;
```

Транспонирует таблицу (функция premiya_dop()) для удобства просмотра данных:

```
SELECT id_worker,FIO,Position, [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11],
[12]
FROM dbo.premiya_dop() PIVOT(sum([Количество услуг])
FOR[q] IN([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12])) AS test_pivot
```

Табличные функции:

premtiya – подсчитывает количество выполненных услуг для каждого работника

```
create function [dbo].[premiya]()
returns table as return
(select dbo.worker.id_worker,FIO,Position,count(dbo.order_service.id_service) as
'Количество услуг'
from dbo.worker join dbo.[order]
on (dbo.worker.id_worker = dbo.[order].id_worker) join dbo.order_service
on (dbo.[order].id_order=dbo.order_service.id_order)
group by dbo.worker.id_worker,FIO,Position)
```

premtiya_dop – подсчитывает количество выполненных услуг для каждого работника по месяцам

```
create function [dbo].[premiya_dop]()
returns table as return
(
```

```

select  dbo.worker.id_worker, FIO, Position, count(dbo.order_service.id_service) as
'Количество услуг',
month(dbo.[order].finish_date) as 'q'
from  dbo.worker join dbo.[order]
on  (dbo.worker.id_worker = dbo.[order].id_worker) join dbo.order_service
on  (dbo.[order].id_order=dbo.order_service.id_order)
group by  dbo.worker.id_worker, FIO, Position, dbo.[order].finish_date

```

price_of – выводит цену для каждой заявки за текущий месяц

```

create function [dbo].[price_of]()
returns table
AS return
(select  id_order, price, order_date from [dbo].[order]
where (MONTH(GETDATE())=MONTH(order_date))

```

Скалярные функции:

TOP_spare – выводит самую используемую деталь за все время

```

create function [dbo].[TOP_spare]()
returns nvarchar(50)
begin
declare @top nvarchar(50)
declare @var int;
declare @max int = 0;
declare @nemax int
declare curs cursor for
select  distinct id_spare_part from dbo.[order_service] join dbo.service
on(dbo.order_service.id_service = dbo.service.id_service)
join dbo.spare_part_service on (dbo.spare_part_service.id_service =
dbo.service.id_service)

open curs
fetch next from curs into @var
while @@FETCH_STATUS=0
begin
select @nemax= (select Count(*) from dbo.[order_service] join dbo.service
on(dbo.order_service.id_service = dbo.service.id_service)
join dbo.spare_part_service on (dbo.spare_part_service.id_service =
dbo.service.id_service)
where id_spare_part =@var)

if (@nemax > @max )
begin
select @top= (select specification from dbo.spares where id_spare_part= @var )
select @max = @nemax
end
fetch next from curs into @var
end
close curs ;
deallocate curs;
return @top
end

```

stringCLR – выводит менее используемую деталь за все время

```

create FUNCTION [dbo].[stringCLR]()
RETURNS [nvarchar](50) WITH EXECUTE AS CALLER
AS
EXTERNAL NAME [CLRFunctions].[UserDefinedFunctions].[SqlFunction1]

```

Код CLR-функции на C#:

```
public static SqlString SqlFunction1()
{
    using (SqlConnection conn = new SqlConnection("Data source =DESKTOP-42QTCGA ;
Initial Catalog = TP; Integrated Security = True"))
    {
        conn.Open();
        string comm;
        comm = "begin " +
            "declare @top nvarchar(50);" +
            "declare @var int;" +
            "declare @max int = 100;" +
            "declare @nemax int;" +
            "declare curs cursor for " +
            "select distinct id_spare_part from dbo.[order_service] join dbo.service " +
            "on(dbo.order_service.id_service = dbo.service.id_service) " +
            "join dbo.spare_part_service on (dbo.spare_part_service.id_service =
dbo.service.id_service) ;" +
            "open curs; " +
            "fetch next from curs into @var;" +
            "while @@FETCH_STATUS=0 " +
            "begin " +
            "select @nemax= (select Count(*) from dbo.[order_service] join
dbo.service " +
            "on(dbo.order_service.id_service = dbo.service.id_service) " +
            "join dbo.spare_part_service on (dbo.spare_part_service.id_service =
dbo.service.id_service) " +
            "where id_spare_part =@var);" +
            "if (@nemax < @max ) " +
            "begin " +
            "select @top= (select specification from dbo.spares where id_spare_part=
@var );" +
            "select @max = @nemax;" +
            "end; " +
            "fetch next from curs into @var;" +
            "end;" +
            "close curs ;" +
            "deallocate curs;" +
            "select @top;" +
            "end;";
        SqlCommand cmd = new SqlCommand(comm, conn);
        conn.Close();
        return (SqlString)cmd.ExecuteScalar();
    }
}
```

Триггеры:

PlusZP – после изменения цены заявки, добавляет 30% от её цены к зарплате работника, который её выполняет

```
create trigger [dbo].[PlusZP] on [dbo].[order]
after update as
begin
declare @price_order int
declare @ZP int
declare @newZP int
select @ZP = (select salary from dbo.Position inner join worker
on (dbo.Position.position = dbo.worker.position) inner join inserted on
(dbo.worker.id_worker= inserted.id_worker) )
select @price_order = (select price from inserted)
select @newZP = @ZP + (@price_order * 0.3)
update dbo.Position set salary = @newZP where dbo.Position.position = (select
dbo.Position.position from dbo.Position inner join worker
```

```
on (dbo.Position.position = dbo.worker.position) inner join inserted on
(dbo.worker.id_worker= inserted.id_worker))
end
```

Plus_price – после добавления услуги к заявке изменяется цена заявки и количество деталей на складе, которые необходимы для предоставления услуги, уменьшается на одну

```
create trigger [dbo].[Plus_price] on [dbo].[order_service]
after insert as
begin
declare @id_ord int;
declare @id_ser int;
declare @price_ser int;
select @id_ord = (select id_order from inserted);
select @id_ser = (select id_service from inserted);
select @price_ser = (select price from dbo.service where id_service=@id_ser);
update dbo.[order] set price += @price_ser where id_order = @id_ord

declare @id_spar int
declare curs CURSOR FOR
select dbo.spare_part_service.id_spare_part from dbo.spare_part_service
where dbo.spare_part_service.id_service =@id_ser

open curs
fetch next from curs into @id_spar
while @@FETCH_STATUS=0

begin
update dbo.spares set quantity = quantity -1 where dbo.spares.id_spare_part = @id_spar
fetch next from curs into @id_spar
end
close curs
deallocate curs
end
```

Процедуры изменения данных (для таблицы client):

del_client(@id_client) – процедура удаления записи из таблицы client

```
create procedure [dbo].[del_client]
(@id_client int)
as
begin
delete from dbo.client where id_client = @id_client
end
```

up_client (@id_client int,@surname nvarchar(50),@name nvarchar(50),@patronymic nvarchar(50),@phone_number nvarchar(50)) – обновление таблицы client

```
create procedure [dbo].[up_client]
(@id_client int,@surname nvarchar(50),@name nvarchar(50),@patronymic
nvarchar(50),@phone_number nvarchar(50))
as
begin
update dbo.client set
surname = @surname,
name = @name,
patronymic = @patronymic,
phone_number = @phone_number
where id_client = @id_client
end
```

add_client (@surname nvarchar(50),@name nvarchar(50),@patronymic nvarchar(50),@phone_number nvarchar(50)) – ввод данных в таблицу client

```

create procedure [dbo].[add_client]
(@surname nvarchar(50),@name nvarchar(50),@patronymic nvarchar(50),@phone_number
nvarchar(50))
as
begin
insert into dbo.client(surname,name,patronymic,phone_number)
values (@surname,@name ,@patronymic,@phone_number)
end

```

4. Создание приложения для работы с базой данных

Программа разрабатывалась в среде Visual Studio на платформе Microsoft .Net Framework. Логика программы и графический интерфейс реализованы на языке C# с использованием встроенного класса Windows Forms.

Для реализации доступа (инженер – ремонтник, менеджер по закупкам, бухгалтер, администратор) создавались отдельные формы.

Описание некоторых функций:

PasswordAll(string id) – функция необходимая для проверки пароля

```

public static string PasswordAll(string id)
{
    string p;
    bool validValue;
    SqlConnection conn = new SqlConnection("Data source =DESKTOP-42QTCGA ;
Initial Catalog = TP; Integrated Security = True");
    conn.Open();

    using (conn.CreateCommand())
    {
        SqlCommand pass = new SqlCommand("select * from TP.dbo.password where
TP.dbo.password.id_pass = " + id, conn);
        using (SqlDataReader dataReader = pass.ExecuteReader())
        {
            validValue = dataReader.Read();
            p = dataReader["pass"].ToString();
        }
    }
    conn.Close();
    return p;}

```

Подключение осуществляется:

```

SqlConnection conn = new SqlConnection(@"Data source =DESKTOP-42QTCGA ; Initial
Catalog = TP; Integrated Security = True");

```

В реализации большинства форм использовались функции:

Txt(string str,string name stolb)- возвращает строку, полученную при запросе

```

public string Txt(string str, string name_stolb)
{
    bool validValue;
    string smm;
    using (conn.CreateCommand())
    {
        SqlCommand com = new SqlCommand(str, conn);
    }
}

```

```

        using (SqlDataReader dataReader = com.ExecuteReader())
        {
            validValue = dataReader.Read();
            smm = dataReader["smm"].ToString();
        }
    }
    return smm;
}

```

Command(string str) – выполняет запрос str и возвращает таблицу типа DataTable

```

public DataTable Command(string str)
{
    try
    {
        using (SqlCommand sql = conn.CreateCommand())
        {
            SqlDataAdapter sqlDA = new SqlDataAdapter(str, conn);
            DataTable dtbl = new DataTable();
            sqlDA.Fill(dtbl);
            return dtbl;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, "ERROR");
        conn.Close();
        return null;
    }
}

```

Chektext(string str) - функция преобразования строки для SQL

```

private string chektext(string str)
{
    if (str == "")
        return null;
    else
        return ("'" + str + "'");
}

```

Рассмотрим функции изменения данных в БД:

Функция вставки в таблицу spares:

```

string spec = chektext(textBox2.Text);
string manuf = chektext(textBox3.Text);
string price = textBox4.Text;
string quantity = textBox5.Text;

Command("exec add_spares " + spec + ", " + manuf + ", " + price + ", " +
quantity );
dataGridView2.DataSource = Command("select * from spares");

```


Функции изменения данных в таблице spares:

Выбирается ячейка и в поле TextBox вводится новое значение, затем нажимается кнопка «Изменить».

Для одного значения в строке:

```
string id;
int row = Convert.ToInt32(dataGridView2.CurrentRow.Index);
int cell = Convert.ToInt32(dataGridView2.CurrentCell.ColumnIndex);
id = dataGridView2.Rows[row].Cells[0].Value.ToString();

//ошибка, если изменять PK
string value = textBox1.Text;
if (cell != 0)
{
    string header = dataGridView2.Columns[cell].HeaderText.ToString();
    string headerPK = dataGridView2.Columns[0].HeaderText.ToString();
    Type a = dataGridView2.Columns[cell].ValueType; //Получить тип столбца
    string aa = a.Name;
    switch (aa)
    {
        case "String":
            value = "'" + value + "'";
            break;
        case "DateTime":
            value = "'" + value + "'";
            break;
    }
}
string zapr = "update spares set " + header + "=" + value + " where " + headerPK + "=" + id;
Command(zapr);
dataGridView2.DataSource = Command("select * from spares");
textBox1.Text = null;
```

Для всех значений в строке:

```
string id;
int row = Convert.ToInt32(dataGridView2.CurrentRow.Index);
id = dataGridView2.Rows[row].Cells[0].Value.ToString();
string spec = chektext(textBox2.Text);
string manuf = chektext(textBox3.Text);
string price = textBox4.Text;
string quantity = textBox5.Text;
Command("exec up_spares" + id + ", " + spec + ", " + manuf + ", " + price + ", " + quantity);
dataGridView2.DataSource = Command("select * from spares");
```

Функция удаления данных из таблицы spares:

Выбирается строка и нажимается кнопка «Удалить».

```
string id;
int row = Convert.ToInt32(dataGridView2.CurrentRow.Index);
id = dataGridView2.Rows[row].Cells[0].Value.ToString();
string zapr = "exec del_spares " + id;
Command(zapr);
dataGridView2.DataSource = Command("select * from spares");
```

5. Скриншоты:

Форма авторизации:

Окно авторизации

☒ Инженер - ремонтник
☐ Менеджер по закупкам
☐ Бухгалтер
☐ Администратор

Пароль:

Войти

Форма инженера - ремонтника:

Form_repairer

Оформленные заявки | Доп. таблицы

Номер заказа	Номер сотрудника	Имя сотрудника	Дата заказа	Дата выполнения	Услуги	Номер детали	Описание детали
12	3	Работник.1	08.12.2019	08.12.2019	Замена динамика	16	Динамик
12	3	Работник.1	08.12.2019	08.12.2019	Замена динамика	16	Динамик
12	3	Работник.1	08.12.2019	08.12.2019	Замена динамика	16	Динамик
12	3	Работник.1	08.12.2019	08.12.2019	Замена экрана 6.5	4	Клей
12	3	Работник.1	08.12.2019	08.12.2019	Замена экрана 6.5	3	Экран 6.5"
12	3	Работник.1	08.12.2019	08.12.2019	Замена экрана 6.5	4	Клей
12	3	Работник.1	08.12.2019	08.12.2019	Замена экрана 6.5	3	Экран 6.5"
12	3	Работник.1	08.12.2019	08.12.2019	Замена экрана 5	1	Экран 5"
12	3	Работник.1	08.12.2019	08.12.2019	Замена экрана 5	4	Клей
12	3	Работник.1	08.12.2019	08.12.2019	Замена экрана 6.5	4	Клей
12	3	Работник.1	08.12.2019	08.12.2019	Замена экрана 6.5	3	Экран 6.5"
6	7	работник.4	25.11.2019	30.11.2019	Замена динамика	16	Динамик
7	5	Работник.2	25.11.2019	25.11.2019	Защитное стекло	8	Защитное стекло
2	3	Работник.1	25.11.2019	25.11.2019	Замена экрана 6.5	4	Клей
2	3	Работник.1	25.11.2019	25.11.2019	Замена экрана 6.5	3	Экран 6.5"
2	3	Работник.1	25.11.2019	25.11.2019	Замена Аккумулятор бол	7	Аккумулятор бол
3	5	Работник.2	25.11.2019	03.12.2019	Замена экрана 5	1	Экран 5"
3	5	Работник.2	25.11.2019	03.12.2019	Замена экрана 5	4	Клей

ID клиента ID работника Дата окончания ID заявки ID услуги

Назад Добавить Добавить

Form_repairer

Оформленные заявки | Доп. таблицы

specification	manufacturer	quantity
Экран 5"	Китай	11
Экран 6.5"	Китай	6
Клей	Россия	22
Аккумулятор мал	Китай	5
Аккумулятор бол	Китай	7
Защитное стекло	Вьетнам	6
Материнская плата	Китай	4
Клавиатура ноут	Китай	2
Видеокарта	Китай	3
Гнездо TypeC	Китай	5
Динамик	Россия	8
fefefefef	fefefefef	54
fefewfef	ufbf	12

Клиенты Работник Услуги Заявки Запчасти

ID клиента ID работника Дата окончания ID заявки ID услуги

Назад Добавить Добавить

Форма бухгалтера:

Form_accountant

Просмотр | Доп. таблицы

	id_worker	FIO	Position	1	2	3	4	5	6	7	8	9	10	11	12
▶	3	Работник1	Ремонтник1											3	7
	5	Работник2	Ремонтник2											1	2
	6	Работник3	Ремонтник3												1
	7	работник4	Ремонтник4											1	
*															

Количество предоставленных услуг

Отсортировать | Отсортировать по месяцам | Прибыль в текущем месяце с каждой заявки

Назад

Форма администратора:

Form_admin

Клиенты | Работники | Запчасти | Услуги | Услуги-Запчасти | Заявки | Услуги-Заявки

	id_order	id_client	id_worker	order_date	finish_date	price
▶	2	1	3	25.11.2019	25.11.2019	5000
	3	2	5	25.11.2019	03.12.2019	6000
	4	3	6	25.11.2019	11.12.2019	3000
	5	3	3	25.10.2019	29.11.2019	2000
	6	4	7	25.11.2019	30.11.2019	7500
	7	5	5	25.11.2019	25.11.2019	5745
	8	1	3	01.12.2019	12.12.2019	10000
	10	1	3	07.12.2019	07.12.2019	2222
	11	1	3	07.12.2019	07.12.2019	0
	12	3	3	08.12.2019	08.12.2019	17600
*						

Описание | Производитель | Цена | Количество

Добавить | Изменить | Изменить все | Удалить

Назад